

EV369763408

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

VPN Enrollment Protocol Gateway

Inventor(s):
Rudolph Balaz
Victor W. Heller
Xiaohong Su
Keith R. Vogel

ATTORNEY'S DOCKET NO. MS1-467USC2

1 **RELATED APPLICATIONS**

2 This is a division of Application No. 09/548,257, filed April 12, 2000,
3 entitled “VPN Enrollment Protocol Gateway”.

4

5 **TECHNICAL FIELD**

6 This invention relates to secure communications, and more particularly to a
7 protocol gateway allowing routers operating in accordance with one protocol to
8 obtain and maintain certificates for a virtual private network (VPN) from a
9 certificate authority operating in accordance with another protocol.

10

11 **BACKGROUND OF THE INVENTION**

12 Computer technology is continually advancing, resulting in continually
13 evolving uses for computers. One such use is communicating with other
14 computers over a network, such as the Internet, to obtain or exchange information,
15 purchase or sell goods or services, etc. One particular type of communication that
16 can be established is referred to as a “virtual private network” or “VPN”. In a
17 VPN, portions of a network (such as the Internet) are used to establish secure
18 communications from one computer to another via multiple different routers in the
19 network. The VPN allows users to use the larger network (e.g., the Internet) to
20 connect to another computer as if they were part of a dedicated secure network.

21 In order to operate as part of a VPN, a router enrolls for a VPN certificate
22 via a certificate authority (CA). This VPN certificate is then provided to other
23 routers that are part of the VPN and is used to authenticate the router and may also
24 be used to securely communicate with the other routers. However, different
25 protocols for enrolling for VPN certificates have arisen, many of which are

1 incompatible with one another. For example, many routers available from Cisco
2 Systems, Inc. of San Jose, California use a proprietary protocol called Simple
3 Certificate Enrollment Protocol (SCEP) for obtaining VPN certificates, while
4 many certificate authorities available from Microsoft Corporation of Redmond,
5 Washington use an incompatible enrollment protocol based on Public-Key
6 Cryptography Standard (PKCS) #10 and PKCS #7. Thus, a router using SCEP
7 would not be able to enroll for a VPN certificate from a CA using PKCS #10 and
8 PKCS #7.

9 Additionally, many routers and CAs are already manufactured and in use
10 that operate based on such incompatible protocols. Therefore, re-designing such
11 routers or CAs to be compatible with one another would require the replacement
12 of many such pre-existing devices. Thus, it would be beneficial to provide a
13 solution that allows routers and CAs (including pre-existing routers and CAs)
14 operating based on incompatible protocols to communicate with one another for
15 VPN certificate enrollment.

16 The VPN enrollment protocol gateway described below addresses these and
17 other disadvantages.

18

19 **SUMMARY OF THE INVENTION**

20 A virtual private network (VPN) enrollment protocol gateway is described
21 herein. The protocol gateway allows routers operating in accordance with one
22 protocol to obtain and maintain certificates for a VPN from a certificate authority
23 operating in accordance with another protocol.

24 According to one aspect, the VPN enrollment protocol gateway is
25 implemented as a registration authority that operates as an intermediary between

1 the router and the certificate authority. As a registration authority, the gateway is
2 trusted by the certificate authority. The router communicates with the registration
3 authority as if it were the certificate authority, not realizing that it is
4 communicating with an intermediary.

5 According to another aspect, the protocol gateway receives a router
6 enrollment request from the router. The protocol gateway decrypts the request,
7 adds an alterative subject name to the request, digitally signs the request, and
8 forwards the signed request to the certificate authority. The certificate authority
9 determines whether to trust the source of the request (the protocol gateway), and
10 proceeds to respond with the requested certificate if it verifies that the gateway can
11 be trusted. The gateway receives the requested certificate, encrypts and digitally
12 signs a response including the certificate, and returns the signed and encrypted
13 response to the router.

14 According to another aspect, the certificate authority may not be able to
15 immediately issue a certificate, in which case it issues a pending response. The
16 registration authority maintains a mapping of a router transaction ID (identifier)
17 received from the router and a pending response ID received from the certificate
18 authority. This mapping allows subsequent requests from the router with the same
19 transaction ID (e.g., querying whether the certificate has been issued yet) to be
20 properly matched to a request previously submitted to the certificate authority for
21 which a pending response was issued. The registration authority also maintains a
22 mapping of a hash value of the request received from the router to the pending
23 response for that request. This mapping allows the registration authority to
24 determine when a request is resubmitted by the router (e.g., in the event the router
25 never receives a pending response returned to it by the registration authority).

1 According to another aspect, the protocol gateway receives a get certificate
2 revocation list from the router. The protocol gateway decrypts the request and
3 extracts from the request the certificate serial number of the signing certificate of
4 the request. The protocol gateway then submits a Get Certificate by Serial
5 Number request to the certificate authority, which returns to the protocol gateway
6 the certificate corresponding to the serial number. The protocol gateway extracts a
7 certificate revocation list distribution point from the response, and obtains the
8 certificate revocation list from the distribution point. The protocol gateway then
9 generates a response including the certificate revocation list, encrypts and signs
10 the response, and returns the response to the router.

11 According to another aspect, the protocol gateway receives a get certificate
12 request from the router. The protocol gateway decrypts the request and extracts
13 from the request the certificate serial number of the signing certificate of the
14 request. The protocol gateway then submits a Get Certificate by Serial Number
15 request to the certificate authority, which returns to the protocol gateway the
16 certificate corresponding to the serial number. The protocol gateway then encrypts
17 and signs a response including the certificate, and returns the response to the
18 router.

19 According to another aspect, the protocol gateway receives a get certificate
20 authority certificate request from the router. The protocol gateway generates a
21 response message including the signing certificate of the registration authority as
22 well as the encryption certificate of the registration authority, and returns the
23 response message to the router.

24 According to another aspect, the protocol gateway maintains a record of
25 passwords handed out to a router. A router obtains a password by communicating

1 with the protocol gateway (or another device trusted by the protocol gateway) via
2 an authenticatable mechanism (e.g., SSL (Secure Sockets Layer)). A password is
3 returned to the router, which can then use this password for a request submitted to
4 the protocol gateway. If the password presented by the router is in the router's
5 record, then the request is processed; otherwise, the request is rejected.

6

7 **BRIEF DESCRIPTION OF THE DRAWINGS**

8 The present invention is illustrated by way of example and not limitation in
9 the figures of the accompanying drawings. The same numbers are used
10 throughout the figures to reference like components and/or features.

11 Fig. 1 shows a virtual private network environment with an enrollment
12 protocol gateway in accordance with certain embodiments of the invention.

13 Fig. 2 shows a general example of a computer that can be used in
14 accordance with certain embodiments of the invention.

15 Fig. 3 is a block diagram illustrating a registration authority operating as a
16 protocol gateway between a router and a certificate authority in accordance with
17 certain embodiments of the invention.

18 Fig. 4 shows an exemplary transaction ID table in accordance with certain
19 embodiments of the invention.

20 Fig. 5 shows an exemplary request hash table in accordance with certain
21 embodiments of the invention.

22 Fig. 6 shows an exemplary password table in accordance with certain
23 embodiments of the invention.

24

25

1 Figs. 7a and 7b are a flowchart illustrating an exemplary process for
2 handling a router enrollment request in accordance with certain embodiments of
3 the invention.

4 Fig. 8 is a flowchart illustrating an exemplary process for handling pending
5 responses in accordance with certain embodiments of the invention.

6 Fig. 9 is a flowchart illustrating an exemplary process for handling a Get
7 Certificate Revocation List request in accordance with certain embodiments of the
8 invention.

9 Fig. 10 is a flowchart illustrating an exemplary process for handling a Get
10 Certificate request in accordance with certain embodiments of the invention.

11 Fig. 11 is a flowchart illustrating an exemplary process for handling a Get
12 Certificate Authority Certificate request in accordance with certain embodiments
13 of the invention.

14 Fig. 12 is a flowchart illustrating an exemplary process for distributing and
15 verifying passwords in accordance with certain embodiments of the invention.

16

17 **DETAILED DESCRIPTION**

18 The discussion herein assumes that the reader is familiar with cryptography.
19 For a basic introduction of cryptography, the reader is directed to a text written by
20 Bruce Schneier and entitled "Applied Cryptography: Protocols, Algorithms, and
21 Source Code in C," published by John Wiley & Sons with copyright 1994 (or
22 second edition with copyright 1996).

23 In the discussion below, embodiments of the invention will be described in
24 the general context of computer-executable instructions, such as program modules,
25 being executed by one or more conventional personal computers. Generally,

1 program modules include routines, programs, objects, components, data structures,
2 etc. that perform particular tasks or implement particular abstract data types.
3 Moreover, those skilled in the art will appreciate that various embodiments of the
4 invention may be practiced with other computer system configurations, including
5 hand-held devices, multiprocessor systems, microprocessor-based or
6 programmable consumer electronics, network PCs, minicomputers, mainframe
7 computers, and the like. In a distributed computer environment, program modules
8 may be located in both local and remote memory storage devices.

9 Alternatively, embodiments of the invention can be implemented in
10 hardware or a combination of hardware, software, and/or firmware. For example,
11 all or part of the invention can be implemented in one or more application specific
12 integrated circuits (ASICs).

13 Fig. 1 shows a virtual private network environment with an enrollment
14 protocol gateway in accordance with certain embodiments of the invention.
15 Generally, one or more client computers 102 can communicate with one or more
16 server computers 104 via a public network supporting a conventional virtual
17 private network (VPN) 106. Server computers 104 can be coupled directly to the
18 network supporting VPN 106, or alternatively can be coupled to the network
19 supporting VPN 106 via another network, such as local area network (LAN) 108.

20 VPN 106 includes one or more routers 110, 112, and 114 through which
21 data is passed between client 102 and server 104. Routers 110 – 114 are part of a
22 public network, such as the Internet. Routers that are part of other types of
23 networks may also be included in VPN 106, such as routers from a LAN or a
24 private wide-area network.

1 Additionally, other networks may be involved in the communication
2 between client 102 and server 104. By way of example, client 102 may connect to
3 the public network supporting VPN 106 via a conventional modem and a Public
4 Switched Telephone Network (PSTN), via a conventional cable modem and cable
5 lines, etc.

6 Routers 110 – 114 can communicate with one another, as well as
7 registration authority 118, via any of a wide variety of conventional
8 communications protocols. In one implementation, routers 110 – 114
9 communicate with one another and registration authority 118 using the Hypertext
10 Transfer Protocol (HTTP).

11 Each of the routers 110 – 114 receives data from one of the other routers
12 110 – 114 or alternatively from another component (e.g., a public network access
13 provider, such as an Internet Service Provider (ISP); client computer 102; etc.).
14 The data is then securely passed on to another of the routers 110 – 114 or other
15 components.

16 In order for data to be transmitted among routers 110 – 114, a certificate-
17 based authentication scheme is employed. In such an authentication scheme, each
18 router 110 – 114 is assigned a unique certificate that it can use to authenticate
19 itself to other routers or other computing devices (e.g., an ISP, a bridge or
20 gateway, etc.). Additionally, these other computing devices may be part of VPN
21 106 and may similarly be assigned unique certificates that can be used for
22 authentication. Such certificates can also optionally be used to encrypt messages
23 between routers and/or other computing devices in any of a variety of
24 conventional manners. For ease of explanation, routers are described as the
25 devices that are obtaining and maintaining certificates for VPN 106. The

1 establishment and operation of a VPN is well-known to those skilled in the art,
2 and thus will not be discussed further except as it pertains to the invention.

3 The certificates used by routers 110 – 114 are assigned by a trusted
4 certificate authority (CA) 116. The process of obtaining such a certificate is
5 referred to as “enrollment”. In the illustrated example, routers 110 – 114 use a
6 different enrollment protocol than is used by certificate authority 116. A
7 registration authority 118 communicates with both routers 110 – 114 and
8 certificate authority 116 and acts as an intermediary for enrollment, translating
9 requests and responses in one protocol to another, as discussed in more detail
10 below.

11 Fig. 2 shows a general example of a computer 142 that can be used in
12 accordance with certain embodiments of the invention. Computer 142 is shown as
13 an example of a computer that can perform the functions of a client computer 102,
14 a server computer 104, a certificate authority 116, or a registration authority 118
15 of Fig. 1. Computer 142 includes one or more processors or processing units 144,
16 a system memory 146, and a bus 148 that couples various system components
17 including the system memory 146 to processors 144.

18 The bus 148 represents one or more of any of several types of bus
19 structures, including a memory bus or memory controller, a peripheral bus, an
20 accelerated graphics port, and a processor or local bus using any of a variety of
21 bus architectures. The system memory includes read only memory (ROM) 150
22 and random access memory (RAM) 152. A basic input/output system (BIOS) 154,
23 containing the basic routines that help to transfer information between elements
24 within computer 142, such as during start-up, is stored in ROM 150. Computer
25 142 further includes a hard disk drive 156 for reading from and writing to a hard

1 disk, not shown, connected to bus 148 via a hard disk driver interface 157 (e.g., a
2 SCSI, ATA, or other type of interface); a magnetic disk drive 158 for reading from
3 and writing to a removable magnetic disk 160, connected to bus 148 via a
4 magnetic disk drive interface 161; and an optical disk drive 162 for reading from
5 or writing to a removable optical disk 164 such as a CD ROM, DVD, or other
6 optical media, connected to bus 148 via an optical drive interface 165. The drives
7 and their associated computer-readable media provide nonvolatile storage of
8 computer readable instructions, data structures, program modules and other data
9 for computer 142. Although the exemplary environment described herein employs
10 a hard disk, a removable magnetic disk 160 and a removable optical disk 164, it
11 should be appreciated by those skilled in the art that other types of computer
12 readable media which can store data that is accessible by a computer, such as
13 magnetic cassettes, flash memory cards, digital video disks, random access
14 memories (RAMs) read only memories (ROM), and the like, may also be used in
15 the exemplary operating environment.

16 A number of program modules may be stored on the hard disk, magnetic
17 disk 160, optical disk 164, ROM 150, or RAM 152, including an operating system
18 170, one or more application programs 172, other program modules 174, and
19 program data 176. Operating system 170 can be any of a variety of operating
20 systems, such as any of the "Windows" family of operating systems available from
21 Microsoft Corporation of Redmond, Washington. A user may enter commands
22 and information into computer 142 through input devices such as keyboard 178
23 and pointing device 180. Other input devices (not shown) may include a
24 microphone, joystick, game pad, satellite dish, scanner, or the like. These and
25 other input devices are connected to the processing unit 144 through an interface

1 168 (e.g., a serial port interface) that is coupled to the system bus. A monitor 184
2 or other type of display device is also connected to the system bus 148 via an
3 interface, such as a video adapter 186. In addition to the monitor, personal
4 computers typically include other peripheral output devices (not shown) such as
5 speakers and printers.

6 Computer 142 can operate in a networked environment using logical
7 connections to one or more remote computers, such as a remote computer 188.
8 The remote computer 188 may be another personal computer, a server, a router, a
9 network PC, a peer device or other common network node, and typically includes
10 many or all of the elements described above relative to computer 142, although
11 only a memory storage device 190 has been illustrated in Fig. 2. The logical
12 connections depicted in Fig. 2 include a local area network (LAN) 192 and a wide
13 area network (WAN) 194. Such networking environments are commonplace in
14 offices, enterprise-wide computer networks, intranets, and the Internet. In the
15 described embodiment of the invention, remote computer 188 executes an Internet
16 Web browser program such as the “Internet Explorer” Web browser manufactured
17 and distributed by Microsoft Corporation of Redmond, Washington.

18 When used in a LAN networking environment, computer 142 is connected
19 to the local network 192 through a network interface or adapter 196. When used
20 in a WAN networking environment, computer 142 typically includes a modem 198
21 or other means for establishing communications over the wide area network 194,
22 such as the Internet. The modem 198, which may be internal or external, is
23 connected to the system bus 148 via a serial port interface 168. In a networked
24 environment, program modules depicted relative to the personal computer 142, or
25 portions thereof, may be stored in the remote memory storage device. It will be

1 appreciated that the network connections shown are exemplary and other means of
2 establishing a communications link between the computers may be used.

3 Generally, the data processors of computer 142 are programmed by means
4 of instructions stored at different times in the various computer-readable storage
5 media of the computer. Programs and operating systems are typically distributed,
6 for example, on floppy disks or CD-ROMs. From there, they are installed or
7 loaded into the secondary memory of a computer. At execution, they are loaded at
8 least partially into the computer's primary electronic memory. The invention
9 described herein includes these and other various types of computer-readable
10 storage media when such media contain instructions or programs for implementing
11 the steps described below in conjunction with a microprocessor or other data
12 processor. The invention also includes the computer itself when programmed
13 according to the methods and techniques described below. Furthermore, certain
14 sub-components of the computer may be programmed to perform the functions
15 and steps described herein. The invention includes such sub-components when
16 they are programmed as described. In addition, the invention described herein
17 includes data structures, described herein, as embodied on various types of
18 memory media.

19 For purposes of illustration, programs and other executable program
20 components such as the operating system are illustrated herein as discrete blocks,
21 although it is recognized that such programs and components reside at various
22 times in different storage components of the computer, and are executed by the
23 data processor(s) of the computer.

24 Fig. 3 is a block diagram illustrating an exemplary registration authority
25 118 operating as a protocol gateway between a router 210 and a certificate

1 authority 116. Router 210 can be, for example, any of routers 110 – 114 of Fig. 1.
2 Router 210 is configured (e.g., during an installation or setup process) with the
3 address of registration authority 118 rather than CA 116 as the certificate
4 authority. In the illustrated example, router 210 has no other knowledge that it is
5 communicating with registration authority 118 rather than certificate authority
6 116.

7 Communication between registration authority 118 and each of router 210
8 and certificate authority 116 can be carried out using any of a wide variety of
9 conventional encryption and/or digital signing techniques. By way of example,
10 using well-known public key cryptography techniques, a device obtains a private
11 key/public key pair; the public key is made available to other devices while the
12 private key is kept secret by the device. Another device can encrypt a message
13 intended for this device by using a conventional encryption algorithm and this
14 device's public key. The private key/public key pair and the encryption algorithm
15 are chosen such that it is relatively easy to decrypt the message with the private
16 key, but extremely difficult to decrypt the message without the private key.
17 Similarly, a message can be digitally signed by the device using a conventional
18 encryption algorithm and its private key. The digitally signed message can be
19 decrypted by another device using the public key, allowing the other device to
20 verify that the message came from that device. Alternatively, rather than applying
21 an encryption algorithm to the message itself, the encryption algorithm may be
22 applied to a hash value generated based on the message and a known hash
23 function. Different public key/private key pairs can be used for encryption and
24 digital signatures, or alternatively the same public key/private key pair can be used
25 for both encryption and digital signatures.

1 Registration authority 118 operates as an enrollment agent for certificate
2 authority 116, allowing routers such as router 210 to enroll for a VPN certificate
3 from certificate authority 116 via registration authority 118. Registration authority
4 118 obtains, from certificate authority 116, an enrollment agency signature
5 certificate (e.g., by enrolling for an “Offline IPSec” enrollment agent signature
6 certificate) and an encryption certificate (e.g., by enrolling for an “IPSec
7 Encryption” certificate). In the illustrated examples, these certificates are used by
8 registration authority 118 to digitally sign data sent to both the router 210 and the
9 certificate authority 116, and to encrypt data sent to the router 210.

10 Router 210 communicates requests 212 to registration authority 118 in
11 accordance with the protocol supported by router 210. In the illustrated example,
12 router 210 supports the protocol SCEP. Different types of requests 212 can be
13 transmitted to registration authority 118. In one implementation, registration
14 authority 118 operates as a protocol gateway for the following types of requests:
15 router enrollment, get certificate revocation list (CRL), get certificate, get
16 certificate authority (CA) certificate, and password registration. The specific
17 manner in which each of these requests is handled by registration authority 118 is
18 discussed in more detail below.

19 Upon receipt of an SCEP request 212, registration authority 118 converts
20 the request into an appropriate format for certificate authority 116. The converted
21 request is then digitally signed by registration authority 118 and the signed request
22 214 is transmitted to certificate authority 116. Certificate authority 116, receiving
23 a request in its own protocol (using PKCS #7 and PKCS #10), responds to the
24 request and issues a CA response 216. Registration authority 118 receives the
25 response 216, converts the response to the appropriate SCEP format for router

1 210, and transmits an SCEP response 218 to router 210. Alternatively, for some
2 requests registration authority 118 may generate the response 218 without
3 forwarding a signed request 214 to certificate authority 116.

4 Registration authority 118 includes a protocol converter 220. Protocol
5 converter 220 receives messages from router 210 and converts them as necessary
6 to the proper protocol for certificate authority 116, and similarly receives
7 messages from certificate authority 116 and converts them to the proper protocol
8 for router 210. The manner in which protocol converter 220 operates is dependent
9 on the particular protocols being used by router 210 and certificate authority 116.

10 In one implementation, registration authority 118 operates in accordance
11 with the Internet X.509 Public Key Infrastructure Certificate and CRL Profile
12 (Network Working Group Request for Comments 2459, January 1999).
13 Alternatively, other implementations may operate in accordance with other
14 standards.

15 Registration authority 118 also includes a transaction ID table 222, a
16 request hash table 224, and a password table 226. Tables 222 – 226 are used by
17 registration authority 118 to maintain information regarding requests 212 and
18 responses 216 in order to conform with the protocols of router 210 and certificate
19 authority 116.

20 Fig. 4 shows an exemplary transaction ID table in accordance with certain
21 embodiments of the invention. Transaction ID table 222 maintains a mapping of
22 router transaction IDs 228 to CA request IDs 230. A router transaction ID 228 is
23 received by registration authority 118 from router 210 as part of each router
24 enrollment message. Similarly, when certificate authority 116 returns a pending
25 response to registration authority 118, the pending response includes a CA request

1 ID 230 (also referred to as a "token"). Transaction ID table 222 allows registration
2 authority 118 to query certificate authority 116 for the correct certificate in
3 response to subsequent requests from router 210 for the certificate the pending
4 response was issued for, as discussed in more detail below.

5 Each entry in transaction ID table 222 is removed from table 222 after a
6 period of time. In one implementation, each entry in table 222 is kept in table 222
7 for one week and then removed. This period of time can optionally be
8 configurable by a user or administrator.

9 Fig. 5 shows an exemplary request hash table in accordance with certain
10 embodiments of the invention. Request hash table 224 maintains a mapping of
11 certificate authority request IDs 232 to hash values of the requests 234. The hash
12 value of a request is generated using any of a variety of conventional hashing
13 functions, such as MD5 (Message Digest 5). A hash function is a mathematical
14 function that, given input data (e.g., the request) generates a unique output hash
15 value based on the input data. Thus, the hash value uniquely identifies a request
16 but requires less storage space than maintaining all of the request. Alternatively,
17 table 224 could maintain the actual request rather than hash values of the request.

18 Request hash table 224 allows registration authority 118 to "remember"
19 router requests. For example, a pending response may be issued by registration
20 authority 118 to router 210, as discussed in more detail below. If a failure or
21 problem occurs during the transmission (e.g., a network failure), then the pending
22 response may not be received by router 210. If router 210 never receives the
23 response, router 210 will re-issue the same request. By maintaining table 224,
24 registration authority 118 can determine when a received request is a re-issued
25

1 request, and need not submit another request for another new certificate to
2 certificate authority 116.

3 Each entry in request hash table 224 is removed from table 224 after a
4 period of time. In one implementation, each entry in table 224 is kept in table 224
5 for twenty minutes and then removed. This period of time can optionally be
6 configurable by a user or administrator.

7 Fig. 6 shows an exemplary password table in accordance with certain
8 embodiments of the invention. Password table 226 maintains passwords 236 that
9 are issued to router 210 in a secure manner. Such passwords can subsequently be
10 used by router 210 to obtain a certificate, providing verification of the identity of
11 router 210.

12 Each password in password table 226 is removed from table 226 after a
13 period of time. In one implementation, each password in table 226 is kept in table
14 226 for sixty minutes and then removed. This period of time can optionally be
15 configurable by an administrator.

16 Returning to Fig. 3, in the illustrated example registration authority 118 is a
17 dynamically linked library (DLL) referred to as the "MSCEP" DLL. Alternatively,
18 registration authority 118 may include a DLL referred to as the "MSCEP" DLL.
19 Registration authority 118 includes a response module 238 that generates
20 responses for certain requests from router 210 that do not require forwarding to
21 certificate authority 116. The operation of response module 238 is discussed in
22 more detail below.

23 Registration authority 118 further hosts a web site 240. Alternatively,
24 registration authority 118 may have a secure communication link to a server
25 hosting web site 240, thereby allowing data to be securely passed between the

1 server and registration authority 118, or registration authority 118 may be software
2 and/or firmware being executed by a server that also hosts web site 240. Web site
3 240 allows passwords to be securely issued to router 210 and stored in password
4 table 226, as discussed in more detail below.

5

6 **Router Enrollment Request**

7 Figs. 7a and 7b are a flowchart illustrating an exemplary process for
8 handling a router enrollment request in accordance with certain embodiments of
9 the invention. Acts on the left-hand side of Figs. 7a and 7b are implemented by
10 registration authority 118 of Fig. 3, while acts on the right-hand side are
11 implemented by certificate authority 116. The process of Figs. 7a and 7b may be
12 performed in software, firmware, hardware, or a combination thereof. Figs. 7a
13 and 7b are described with additional reference to components in Fig. 3.

14 To participate in a VPN, router 210 enrolls for a certificate from certificate
15 authority 116. Router 210 enrolls for a certificate by sending, as SCEP request
16 212, a router enrollment message (e.g., a SCEP PKCSReq message) to registration
17 authority 118. The router enrollment message includes a certificate enrollment
18 request in accordance with the Public-Key Cryptography Standards (PKCS) #10
19 standard. The certificate enrollment request is further encrypted (e.g., using the
20 public key of registration authority 118) and then digitally signed by router 210 in
21 accordance with the Public-Key Cryptography Standards (PKCS) #7 standard.
22 Additional information regarding PKCS #7 and PKCS #10 is available from RSA
23 Data Security, Inc. of Bedford, MA. It should be noted that, although requests
24 from router 210 use PKCS #7 and PKCS #10, certain information needed by
25

1 certificate authority 116 is not included in the requests. Registration authority 118
2 resolves this problem, adding information when necessary.

3 Registration authority 118 receives, as the router enrollment message, this
4 encrypted and digitally signed request (act 242). Upon receipt of the enrollment
5 message, registration authority 118 verifies the signature of the router enrollment
6 message (act 244). If the signature is not verified then the message is ignored (act
7 246). Alternatively, an indication of failure could be returned to router 210.

8 If the signature is verified, then registration authority 118 decrypts the
9 router enrollment message (e.g., using the private key of registration authority
10 118) and extracts the certificate enrollment request from the message (act 248).
11 Registration authority 118 uses the certificate enrollment request to generate a
12 request to the CA for an enrollment certificate in a format expected by certificate
13 authority 116 (act 250).

14 Router 210 needs a certificate with a subject alternative names extension
15 (SubjectAltName). However, router 210 does not specifically request the
16 SubjectAltName extension, and certificate authority 116 does not automatically
17 add the extension. Registration authority 118 resolves this issue by adding, to the
18 message it transmits to certificate authority 116, the SubjectAltName extension in
19 the request.

20 The PKCS #7 message, including both the subject alternative names
21 extension and the certificate enrollment request extracted from the router
22 enrollment message, is digitally signed by registration authority 118 (act 252).
23 This signed message is then transmitted to certificate authority 116 as a CA
24 request (act 254). Note that the CA request thus includes a PKCS #7 message that

1 is signed by registration authority 118, which in turn includes a certificate
2 enrollment request that is signed by router 210.

3 Certificate authority 116 receives the CA request from registration authority
4 118 (act 256) and determines, based on the content of the CA request, whether to
5 issue the requested certificate (act 258). The manner in which certificate authority
6 116 determines whether to issue the requested certificate can vary. In one
7 implementation, certificate authority 116 determines whether to issue a certificate
8 based on whether the certificate of the registration authority 118 can be validated
9 up to a trusted valid root and whether the certificate of registration authority 118
10 includes an extended key usage indicating that registration authority 118 can be a
11 registration authority (and thus operate as an enrollment agent). If both of these
12 conditions are satisfied, then a certificate is issued. Otherwise, the certificate is
13 not issued. Additionally, certificate authority 116 may require that the certificate
14 of registration authority 118 have been issued directly by a certificate authority
15 (that is, no intermediate certificates in the chain from the registration authority
16 certificate to the certificate authority certificate).

17 If certificate authority 116 determines it will not issue a certificate, then
18 certificate authority 116 generates a CA response indicating failure (act 260).
19 However, if certificate authority 116 determines it will issue a certificate, then
20 certificate authority 116 generates the requested certificate (act 262) and then
21 generates a CA response including the generated certificate (act 264).

22 The CA response generated by certificate authority 116 has no message
23 content and is referred to as a "degenerated PKCS #7". The PKCS #7 message,
24 however, allows multiple certificates to be included in a degenerated PKCS #7
25 message. Certificate authority 116 returns the newly generated certificate as part

1 of the degenerated PKCS #7 message. Additionally, the entire certificate chain
2 from the generated certificate up to a root certificate may optionally be included in
3 the degenerated PKCS #7 message.

4 Certificate authority 116 then transmits the CA response (indicating either
5 failure or with the generated certificate) to registration authority 118 (act 266).
6 Registration authority 118 receives the CA response (act 268) and checks whether
7 the CA response includes a certificate (act 270). If no certificate is included, then
8 registration authority 118 generates an SCEP response message indicating failure
9 (act 272). However, if such a certificate is included, then registration authority
10 118 extracts the certificate (act 274) and generates an SCEP response including the
11 certificate (act 276). In the illustrated example, registration authority 118 extracts
12 only the certificate generated by certificate authority 116; the additional certificate
13 chain (if included) is not used by registration authority 118. Alternatively, the
14 entire certificate chain could be included if router 210 desired (or at least could
15 handle) the chain.

16 Registration authority 118 then encrypts the SCEP response (act 278) and
17 digitally signs the encrypted response (act 280). The encrypted and signed
18 response is then transmitted to router 210 (act 282), which in turn can verify the
19 signature and decrypt the response to extract the certificate generated by certificate
20 authority 116.

21

22 **Pending Response Handling**

23 In some situations, certificate authority 116 may not immediately issue a
24 CA response with either a certificate or an indication that no certificate will be
25 issued. For example, certificate authority 116 may wait for an administrator to

1 approve the issuing of the certificate. In such situations, certificate authority 116
2 issues a CA pending response from certificate authority 116.

3 Fig. 8 is a flowchart illustrating an exemplary process for handling pending
4 responses in accordance with certain embodiments of the invention. The process
5 of Fig. 8 is implemented by registration authority 118 of Fig. 3, and may be
6 performed in software, firmware, hardware, or a combination thereof. Fig. 8 is
7 described with additional reference to components in Figs. 3 - 7b.

8 Registration authority 118 receives the CA pending response from
9 certificate authority 116 (act 302). Upon receipt of the CA pending response,
10 registration authority 118 adds entries to its transaction ID table 222 (act 304) and
11 its request hash table 224 (act 306). Registration authority 118 also generates an
12 encrypted and digitally signed SCEP pending response message (act 308) and
13 transmits the encrypted and signed message to router 210 (act 310).

14 Typically, in response to an SCEP pending response message, router 210
15 will re-issue its request for a certificate (e.g., via a GetCertInitial message).
16 Registration authority 118 waits until it receives an additional SCEP request for
17 the certificate from the router 210 (act 312). Once the additional request is
18 received, registration authority 118 accesses transaction ID table 222 to determine
19 the appropriate CA request ID (act 314). Registration authority 118 uses the CA
20 request ID from table 222 to generate a CA request for a certificate corresponding
21 to the CA request ID and digitally signs the CA request (act 316). The signed CA
22 request is then transmitted to certificate authority 116 (act 318).

23 Upon receiving the CA request, certificate authority 116 may issue another
24 pending response to registration authority 118 or alternatively determine whether
25 to issue the certificate (per act 258 of Fig. 7a discussed above). Upon receipt of a

1 response from certificate authority 116, registration authority 118 determines
2 whether the response is another pending response (act 320). If the response is
3 another pending response, the registration authority 118 returns to act 308 and
4 generates and encrypted and signed SCEP pending response message. However, if
5 the response is not another pending response, then registration authority 118
6 proceeds per acts 268 – 282 of Fig. 7b to return an appropriate response to router
7 210.

8 Use of request hash table 224 further allows registration authority 118 to
9 gracefully recover in the event the SCEP pending response message is not received
10 by router 210. If router 210 does not receive the pending response message, then
11 it will resubmit its original request (e.g., an SCEP PKCSReq message). In order to
12 avoid a duplicate request to certificate authority for the certificate, registration
13 authority 118 generates the hash value for SCEP PKCSReq messages it receives
14 and compares the hash value to the entries in request hash table 224. If the hash
15 value matches an entry, then registration authority 118 uses the CA request ID
16 from table 224 to generate a CA request for a certificate corresponding to the CA
17 request ID (act 316), rather than generating a CA request including a certificate
18 enrollment request (act 250 of Fig. 7a). Processing then continues as discussed
19 above with reference to Fig. 8.

20

21 Get Certificate Revocation List Request

22 Returning to Fig. 3, router 210 may also send a Get Certificate Revocation
23 List (CRL) request as SCEP request 212. The request identifies a serial number or
24 similar identifier of a certificate for which the corresponding CRL should be
25 retrieved. The CRL is a list identifying revoked certificates which is made

1 available by the certificate authority (typically in a public repository). The CRL
2 can be checked to determine whether a particular serial number (typically
3 identified in the CRL by its serial number) has been revoked. Registration
4 authority 118 responds to such a request by obtaining the requested CRL and
5 returning it to router 210.

6 Fig. 9 is a flowchart illustrating an exemplary process for handling a Get
7 Certificate Revocation List request in accordance with certain embodiments of the
8 invention. The process of Fig. 9 is implemented by registration authority 118 of
9 Fig. 3, and may be performed in software, firmware, hardware, or a combination
10 thereof. Fig. 9 is described with additional reference to components in Fig. 3.

11 Initially, registration authority 118 receives the Get CRL request (e.g., an
12 SCEP GetCRL message) from router 210 (act 330). Registration authority 118
13 decrypts the request (act 332), verifies the signature of the decrypted request (act
14 334), and proceeds based on whether the signature is verified (act 336). If the
15 signature cannot be successfully verified, then the message is dropped (act 338);
16 registration authority 118 simply ignores the message. Alternatively, registration
17 authority 118 may return an indication to router 210 that the signature could not be
18 verified.

19 However, if the signature is successfully verified, then registration authority
20 118 extracts the certificate serial number from the decrypted request (act 340).
21 This serial number can be extracted by obtaining the serial number of the
22 certificate used by router 210 to sign the Get CRL request.

23 Registration authority 118 then uses the extracted serial number to generate
24 a Get Certificate by Serial Number request (act 342). The Get Certificate by Serial
25 Number request is then digitally signed and transmitted to certificate authority 116

1 (act 344), which in turn accesses its records to identify the certificate
2 corresponding to the given serial number. This certificate is then returned by
3 certificate authority 116 to registration authority 118 (act 346).

4 The certificate returned by certificate authority 116 includes a CRL
5 distribution point, which is an identifier of a location (e.g., a uniform resource
6 locator (URL)) at which the CRL corresponding to the certificate can be obtained.
7 Upon receipt of the certificate, registration authority 118 extracts the CRL
8 distribution point from the certificate (act 348). Registration authority 118 then
9 accesses (e.g., via HTTP) the identified location and retrieves the CRL located
10 there (act 350).

11 Upon obtaining the CRL, registration authority 118 generates an SCEP
12 response message including the CRL (act 352). Registration authority 118 then
13 encrypts and digitally signs the SCEP response message including the CRL, and
14 returns the encrypted and signed SCEP response message to router 210 (act 354).

15 Alternatively, the Get CRL request received from router 210 (act 330) may
16 include the certificate for which the corresponding CRL is to be obtained. In this
17 situation, the CRL distribution point can be extracted by accessing the included
18 certificate, thereby alleviating the need to access certificate authority 116 (acts 340
19 – 346).

20

21 **Get Certificate Request**

22 Returning to Fig. 3, router 210 may also send a Get Certificate request as
23 SCEP request 212. The request identifies a serial number of a certificate that the
24 router would like returned to it. Router 210 may make such a request, for
25 example, in situations where it has kept the serial number of a certificate it needs

1 but has not kept the actual certificate. Registration authority 118 responds to such
2 a request by obtaining the requested certificate and returning it to router 210.

3 Fig. 10 is a flowchart illustrating an exemplary process for handling a Get
4 Certificate request in accordance with certain embodiments of the invention. The
5 process of Fig. 10 is implemented by registration authority 118 of Fig. 3, and may
6 be performed in software, firmware, hardware, or a combination thereof. Fig. 10
7 is described with additional reference to components in Fig. 3.

8 Initially, registration authority 118 receives the Get Certificate request (e.g.,
9 an SCEP GetCert message) from router 210 (act 362). Registration authority 118
10 decrypts the request (act 364), verifies the signature of the decrypted request (act
11 366), and proceeds based on whether the signature is verified (act 368). If the
12 signature cannot be successfully verified, then the message is dropped (act 370);
13 registration authority 118 simply ignores the message. Alternatively, registration
14 authority 118 may return an indication to router 210 that the signature could not be
15 verified.

16 However, if the signature is successfully verified, then registration authority
17 118 extracts the certificate serial number from the decrypted request (act 372).
18 This serial number can be extracted by obtaining the serial number specified in the
19 request (e.g., as the certificate serial number of the signing certificate of the
20 request).

21 Registration authority 118 then uses the extracted serial number to generate
22 a Get Certificate by Serial Number request (act 374). The Get Certificate by Serial
23 Number request is then digitally signed and transmitted to certificate authority 116
24 (act 376), which in turn accesses its records to identify the certificate
25

1 corresponding to the given serial number. This certificate is then returned by
2 certificate authority 116 to registration authority 118 (act 378).

3 Registration authority 118 then generates an SCEP response message
4 including the certificate received in act 378 (act 380). Registration authority 118
5 then encrypts and digitally signs the SCEP response message including the
6 certificate, and returns the encrypted and signed SCEP response message to router
7 210 (act 382).

8

9 **Get CA Request**

10 Returning to Fig. 3, router 210 may also send a Get CA request as SCEP
11 request 212. The request is an HTTP Get call to a URL hosted by registration
12 authority 118. The URL is made available to router 210 during setup or
13 configuration of router 210. Registration authority 118 responds to such a request
14 by returning the requested certificates to router 210.

15 Fig. 11 is a flowchart illustrating an exemplary process for handling a Get
16 Certificate Authority Certificate request in accordance with certain embodiments
17 of the invention. The process of Fig. 11 is implemented by registration authority
18 118 of Fig. 3, and may be performed in software, firmware, hardware, or a
19 combination thereof. Fig. 11 is described with additional reference to components
20 in Fig. 3.

21 Initially, a Get CA request is received by registration authority 118 from
22 router 210 (act 400). Upon receipt of the request, registration authority 118
23 obtains a DLL name identified by the request (act 402). In one implementation, an
24 exemplary Get CA request from router 210 is in the following form:

1 GET mscep.dll/cgi-bin/pkiclient.exe?operation=GetCACert&message=
2 <Base64 encoded authority issuer identifier>

3 In this implementation, registration authority 118 is implemented as an IIS
4 (Internet Information Server) ISAPI (Internet Server Application Programming
5 Interface) DLL. Upon receipt of such a request, IIS parses the input through to
6 identify the first DLL and attempts to load that DLL if necessary. Thus, the
7 remainder of the request can be ignored by registration authority 118 in
determining how to respond to the request.

8 Registration authority 118 is the identified DLL, which in the illustrated
9 example is “mscep.dll”, and passes the request to response module 238 (act 404).
10 In response to being passed the message (either in its entirety, or a part thereof),
11 response module 238 generates a degenerated PKCS #7 message including the
12 signing certificate and the encryption certificate of registration authority 118 (act
13 406), and returns the degenerated PKCS #7 message to the router (act 408). Thus,
14 router 210 requests the certificates for the certificate authority, but receives the
15 certificates for the registration authority instead.

16 Alternatively, registration authority 118 may include a certificate chain in
17 the message it generates in act 408. By way of example, MSCEP DLL 328 may
18 send a certificate request to certificate authority 116, which returns the certificate
19 of certificate authority 116 and a certificate chain that extends up to its root
20 certificate.

21

22 **Password Handling**

23 Returning to Fig. 3, router 210 may also make use of a password to
24 authenticate itself to certificate authority 116 (actually registration authority 118,
25 but router 210 is not aware of this). The password allows registration authority

1 118 (and thus certificate authority 116, which trusts registration authority 118) to
2 know that a particular request actually came from the router claiming to have sent
3 it. The password may be used with one or more of the different types of SCEP
4 requests 212 discussed above. By way of example, the password may be used
5 with the router enrollment request.

6 Fig. 12 is a flowchart illustrating an exemplary process for distributing and
7 verifying passwords in accordance with certain embodiments of the invention.
8 The process of Fig. 12 is implemented by registration authority 118 of Fig. 3, and
9 may be performed in software, firmware, hardware, or a combination thereof. Fig.
10 12 is described with additional reference to components in Fig. 3.

11 Initially, registration authority 118 receives a request for a password (act
12 430). This request is received via a mechanism that allows registration authority
13 118 to authenticate the requestor, such as by use of SSL (Secure Sockets Layer) to
14 authenticate the requestor when accessing web site 240 of Fig. 3. The requestor
15 could be a computer being operated by a router administrator, or alternatively
16 router 210. Upon receipt of the request, registration authority 118 attempts to
17 authenticate the requestor, such as the router administrator, (act 432) and proceeds
18 based on whether the authentication is successful (act 434). If the requestor
19 cannot be authenticated, then the request for a password is denied (act 436). The
20 request may simply be ignored, or alternatively an indication may be returned to
21 the requestor that the request for a password is denied.

22 However, if the router is authenticated, then registration authority 118
23 proceeds to generate a password and add the newly generated password to
24 password table 226 (act 438). The password can be generated by registration
25 authority 118 in any of a wide variety of conventional manners, such as by

1 generating a random (or pseudo-random) number and/or sequence of letters. The
2 generated number may then be placed into a particular format if needed by either
3 router 210 or certificate authority 116, such as hexadecimal format, binary coded
4 decimal format, etc.

5 The password added to password table 226 is removed from table 226 after
6 a period of time. In one implementation, each password in table 226 is kept in
7 table 226 for sixty minutes and then removed. This period of time can optionally
8 be configurable by an administrator.

9 Registration authority 118 then returns the newly generated password to
10 requestor (act 440). This return of the password is done in a secure manner, such
11 as by use of SSL.

12 Eventually, registration authority 118 receives a request from router 210
13 that includes a password that needs to be verified (act 442). Upon receipt of such
14 a request, registration authority 118 determines whether the received password is
15 in password table 226 (act 444). If the received password is not in password table
16 226, then the request is rejected (act 446). The request can simply be ignored, or
17 alternatively a rejection response can be returned to router 210 (e.g., informing
18 router 210 that the password it provided was not valid).

19 However, if the password is in password table 226, then the request is
20 processed by registration authority 118 (act 448). Registration authority 118 may
21 also optionally remove the password from password table 226 (act 450), thereby
22 adding an additional level of security by allowing each password to be used only
23 once.

24
25

1 **Conclusion**

2 Thus, a VPN enrollment protocol gateway has been described. The
3 protocol gateway is implemented as a registration authority that is trusted by the
4 certificate authority, and operates as an intermediary between the router and the
5 certificate authority. The protocol gateway advantageously allows routers
6 operating in accordance with one protocol to obtain and maintain certificates for a
7 VPN from a certificate authority operating in accordance with another protocol.

8 Although the description above uses language that is specific to structural
9 features and/or methodological acts, it is to be understood that the invention
10 defined in the appended claims is not limited to the specific features or acts
11 described. Rather, the specific features and acts are disclosed as exemplary forms
12 of implementing the invention.

13
14
15
16
17
18
19
20
21
22
23
24
25